

CS 595: Assignment #2

Due on Thursday, September 25, 2014

Dr Nelson 4::20PM

Victor Nwala

Contents

Problem 1	3
Problem 2	7
Problem 3	12
OBSERVATIONS AND CONCLUSIONS	20

Problem 1

1. Write a Python program that extracts 1000 unique links from Twitter.

Listing 1: twitter.py

```

from tweepy.streaming import StreamListener
from tweepy import OAuthHandler
from tweepy import Stream
import time
5 import re
import urllib2
import httpplib

#Variables that contains the user credentials to access Twitter API
10 access_token = "384946837-aPnqh9DAtoK1jCSHMepwPJVg27dROGGysYuy9xog"
access_token_secret = "ow458SMzbIcAVZ3RL2nypCGYuqmkaohTNlbZCVBiHG6FC"
consumer_key = "c3SExFZ3K6Do6Yw2Kwi84Str1"
consumer_secret = "CXobLgtDn8feYInLs659BxsjnBTCgfpMD5eEyENilBu6ttbfau"

15 saveFile = open('Freshurls.txt','a')
class StdOutListener(StreamListener):

    def on_data(self, data):
        try:
20
            url = data.split('","url":') [1].split(' "') [0]
            url = url.replace("\\", "/")
            print url

25

            saveFile.write(url)
            saveFile.write('\n')
30 time.sleep(15)
            return True
        except BaseException, e:
            print 'failed ondata,',str(e)

35

    def on_error(self, status):
        print status
        saveFile.close()

40 auth = OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
stream = Stream(auth, StdOutListener())
stream.filter(track=["football"])
saveFile.close()

```

This script extracts random urls from twitter related to the topic inside the filter at the end of the code. I used several filters such as music, movies, football. The URIs extracted may be duplicates or even nonexistent. I wrote two other scripts to correct this.

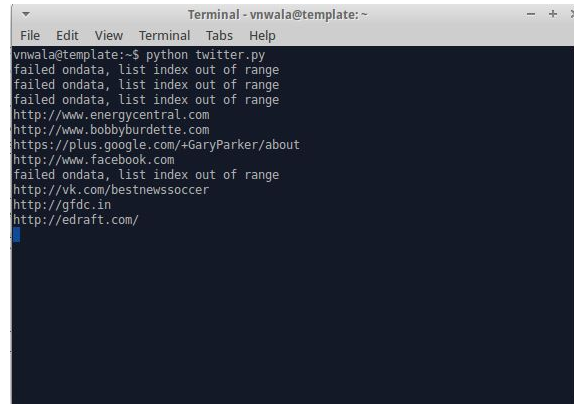


Figure 1: Screen Shot of twitter.py at work

Listing 2: preProcess.py

```

import requests
import urllib2
from urlparse import urlparse
fh = open("Freshurls.txt", 'r')
5 saveFile = open("OKAYUrl.txt", 'a')
for line in fh:
    url=line
    #url=url.lower()

10
    try:
        def get_redirected_url(url):

            opener = urllib2.build_opener(urllib2.HTTPRedirectHandler)
            request = opener.open(url)
15             return request.url
        k =get_redirected_url(url)
        #i = urlparse(k)
        #i =i.netloc
        #i="http://" +i

20

        print k
        saveFile.write(k)
25         saveFile.write('\n')

    except BaseException, e:
        print e
30 saveFile.close()
fh.close()

```

This script above opens up every link extracted from a file and also ensures they are redirected to their final destination before writing them to another file. Hence the final output contains valid URIs

Listing 3: processed.py

```

Terminal - vnwala@template: ~
File Edit View Terminal Tabs Help
vnwala@template:~$ python preProcess.py
http://woodyds.mylvi.net/profile/index.html
http://www.thepensation.com
http://www.podcastgarden.com/podcast/metal4brains
http://bleacherreport.com
http://www.campbellrivermirror.com
http://bronxpinstripes.com/author/bxbombers27/
HTTP Error 400: Bad Request
http://ask.fm/timothydunn00
HTTP Error 404: Not Found
HTTP Error 500: Server Error
http://bleacherreport.com
http://www.dickinsonathletics.com/sports/mbkb/index
http://mestallafeed.nonissue.com/
<urllopen error [Errno -2] Name or service not known>
http://marvel.com
HTTP Error 400: Bad Request
http://inadeportfolio.tumblr.com
HTTP Error 400: Bad Request
http://bleacherreport.com
http://lord-lila.tumblr.com
http://philadelphia.cbslocal.com/

```

Figure 2: Screen Shot of preProcess.py at work

```

Terminal - vnwala@template: ~
File Edit View Terminal Tabs Help
.com/site/error', 'http://www.youtube.com/user/iadorewomen1', 'http://stilvinski
.tumblr.com/# = ', 'https://twitter.com/cinema_posters/status/513425928212754432
/photo/1', 'http://pinkmario.tumblr.com/', 'http://karabomokgoko.tumblr.com', 'h
https://www.youtube.com/watch?v=atlws18rxvs', 'https://twitter.com/ambar_aleman/s
tatus/513425963724840962/photo/1', 'http://www.cracked.com', 'http://illalwaysbe
sixfeetunderthestars.tumblr.com/', 'http://instagram.com/nevaemarine', 'http://c
ristinayanq.tumblr.com', 'http://instagram.com/nochasexo', 'http://www.youtube
.com/?app=desktop', 'https://twitter.com/sikhyaent', 'https://twitter.com/cameron
dallas/status/491973875832324098', 'http://www.askaboutme.com', 'http://iqbaleh.
tumblr.com', 'http://www.amazon.com/gp/registry/wishlist/2o54vidn0q6jd/ref=cm_sw
_su_w', 'http://instagram.com/cmeat', 'http://couturefrommars.tumblr.com', 'htt
p://dimeeknows.tumblr.com', 'http://www.hudl.com/athlete/3505198/highlights/1270
41382', 'https://twitter.com/weare90skids/status/511358870951059456/photo/1', 'h
http://www.cityoftyler.org', 'http://bouledesuiff.flavors.me', 'https://twitter.c
om/dxrlingniall/status/478338992531062784', 'http://www.pinterest.com/giaainpink
', 'https://twitter.com/Louis_tomlinson/status/120620074301267968', 'https://tw
itter.com/', 'http://notoriousjayclyde.tumblr.com/', 'http://savinews.tumblr.co
m/', 'http://geordieoffshore.blogspot.de/', 'http://classic.vb-faq.de/', 'http://
www.cinemalaya.org/', 'http://mentallitch.com', 'https://twitter.com/markzer1021
/status/513426231359011072/photo/1', 'http://instagram.com/kinzeematics', 'http:
//www.youtube.com/user/algemarroncelli', 'http://some-grace.tumblr.com', 'https:
//twitter.com/twerkforbiebsx3/status/513164389000966144/photo/1', 'https://twitt
er.com/groupsexsex8/status/513426255442898945/photo/1', 'http://thenextweb.com',
'https://twitter.com/tbhjoe', 'http://instagram.com/craziestsex', 'http://ask.f

```

Figure 3: Screen Shot of processed.py at work

```
def isInside(arrayOfItems, entry):  
    for line in arrayOfItems:  
        if( line == entry ):  
            return True  
5     return False  
  
inputFileForDuplicates = open("OKAYUrl.txt", "r")  
inputFileArray = inputFileForDuplicates.readlines()  
  
10  
  
outputFileForNonDuplicates = open('NonDup.txt', 'w')  
outputFileArray = []  
  
15  
  
for line in inputFileArray:  
  
    line = line.strip() #remove spaces at the end  
    line = line.lower() #convert to lowercase  
20    #print line  
  
    if( isInside(outputFileArray, line) == False ):  
        outputFileArray.append(line)  
    print outputFileArray  
  
25  
  
outputFileForNonDuplicates.write('\n'.join(str(line) for line in outputFileArray))
```

This script above extracts only unique URIs from the preprocessed file and outputs the final result in another file, hence we have unique and valid URIs.

Problem 2

2. Download the TimeMaps for each of the target URIs. We'll use the mementoweb.org Aggregator. Create a histogram of URIs vs. number of Mementos (as computed from the TimeMaps). For example, 100 URIs with 0 Mementos, 300 URIs with 1 Memento, 400 URIs with 2 Mementos, etc.

Listing 4: countMementos.py

```
import commands
import os, sys

globalMementoUrlDateTimeDelimiter = "*****"
5 def getMementosPages(url):

    if (len(url)>0):

        pages = []
        timemapCount = 0
        timemapPrefix = 'http://mementoproxy.cs.odu.edu/aggr/timemap/link/1/' + url
        while( True ):

            co = 'curl --silent ' + timemapPrefix
            15 page = commands.getoutput(co)
            pages.append(page)

            indexOfRelTimemapMarker = page.rfind('>;rel="timemap"')

            20 if( indexOfRelTimemapMarker == -1 ):
                break
            else:
                #retrieve next timemap for next page of mementos e.g retrieve url
                from <http://mementoproxy.cs.odu.edu/aggr/timemap/link/10001/
                http://www.cnn.com>;rel="timemap"
                i = indexOfRelTimemapMarker -1
                25 timemapPrefix = ''
                while( i > -1 ):
                    if (page[i] != '<'):
                        timemapPrefix = page[i] + timemapPrefix
                    else:
                        30 break
                    i = i - 1

            35 return pages
        else:
            print "url length = 0"

def getItemGivenSignature(page):
    40 if( len(page) > 0 ):
        splitPages0 = page.split(' , <')

        listOfItems = []
        45
```

```

    for i in range(1, len(splitPages0)):

        #splitPagesAgain[url,rel,datetime]
50     if( splitPages0[i].find(';') > -1 ):
        splitPagesAgain = splitPages0[i].split(';')

        #memento signature
        if( splitPagesAgain[1] == 'rel="memento"' ):
55
            url = splitPagesAgain[0]
            url = url[0:len(url)-1]

            if( len(splitPagesAgain)>2 ):
                if( splitPagesAgain[2].find(' datetime="')> -1 ):
                    date = splitPagesAgain[2].strip(' datetime="')
                    date = date[0:len(date)-2]

60
                #print url , globalMementoUrlDateTimeDelimiter, date
                listOfItems.append(url + globalMementoUrlDateTimeDelimiter +
                                    date)

    return listOfItems

70 def countMementos(url):

    if( len(url) > 0 ):
        pages = getMementosPages(url)

75
        countOfMementos = 0
        for i in range(0, len(pages)):
            mementos = getItemGivenSignature(pages[i])

            dummyList = []
80
            if( type(mementos) == type(dummyList) ):
                countOfMementos = countOfMementos + len(mementos)

        return countOfMementos
85
    else:
        return -1

def getMentosCountForFile(inputFilename, outputFilename):
90
    if( len(inputFilename) > 0 and len(outputFilename) > 0 ):

        try:
            inputFile = open(inputFilename, 'r')
            inputUrlsList = inputFile.readlines()
95
            inputFile.close()

            if( len(inputUrlsList) > 0 ):

```



```

        outputFile = open(outputFilename, 'a')

100     except:
        inputFile.close()
        outputFile.close()
        exc_type, exc_obj, exc_tb = sys.exc_info()
        fname = os.path.split(exc_tb.tb_frame.f_code.co_filename)[1]
105     print(fname, exc_tb.tb_lineno, sys.exc_info() )
        return

    #for i in range(726, 0, -1):
110    #for i in range(len(inputUrlsList)-1, 0, -1):
    for i in range(0, len(inputUrlsList)):
        #for i in range(0, 5):
            url = inputUrlsList[i].strip()
            mementosCount = countMementos(url)

115            stringToWrite = url + ', ' + str(mementosCount) + '\n'
            outputFile.write(stringToWrite)
            print i, stringToWrite

120    outputFile.close()

getMementosCountForFile('NonDuplicate.txt', 'outputFileName.txt')

```

To answer this question specifically, I should state that I collected the code from a colleague of mine Alexander Nwala. Hence the code (countMementos.py) belongs to Alexander Nwala but has been modified to suit my purpose. The basic function of this code is to take each URI from my input file, download all its archived versions and output the count to an Output file.

Listing 5: Statistics of Mementos from 1000 URIs

```

URI  MEMENTO
943  0
3    1
1    10
5    109
1    11
1    116
1    123
1    13
10   146
1    15
2    17
2    19
1    196
15   2
1    235
1    24578
1    2469
2    249
20   1    2692

```

	4	3
	1	32
	1	3238
	1	336
25	1	34927
	1	357
	1	385
	2	4
	1	4053
30	1	417
	1	5
	1	52
	1	5291
	1	54
35	1	6
	1	62
	1	66534
	1	68206
	3	7
40	2	71
	1	72
	1	73
	1	8
	1	88
45	1	92

```
Terminal - vnwala@template: ~  
File Edit View Terminal Tabs Help  
vnwala@template:~$ python countMementos.py  
0 https://www.netflix.com/entrytrap?locale=en-us, 19  
1 http://ask.fm/imsohipster95, 0  
2 https://www.goodreads.com/author/show/5805766.thor_garcia/blog, 0  
3 https://twitter.com/taurusthemind_/status/513425650032840704/photo/1, 0  
4 http://sunshinehydrangeas.tumblr.com, 0  
5 http://instagram.com/realadamdeacon, 0  
6 http://newsdigg.net/celebwatch/lindsay_lohan/rss/, 3  
7 http://www.imdb.com/name/nm0000248/, 88  
8 http://www.sweetsourdeals.com, 92  
9 http://fashionablycrohns.blogspot.co.uk, 0
```

Figure 4: Screen Shot of countMementos.py at work

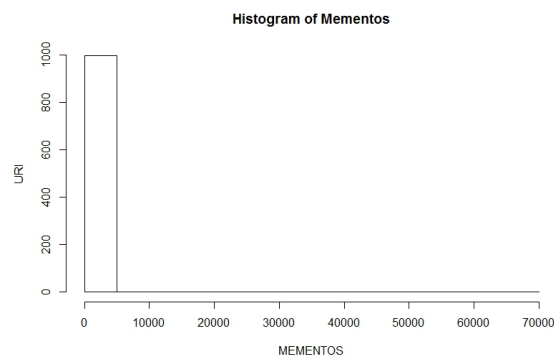


Figure 5: Screen Shot of histogram From Rstudio

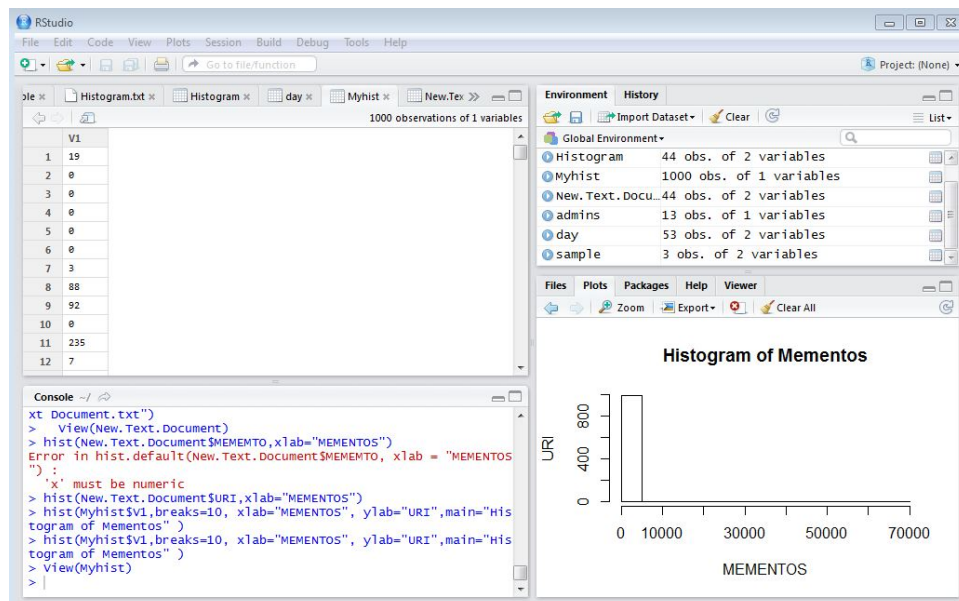


Figure 6: Screen Shot of RstudioLayout

Problem 3

3. Estimate the age of each of the 1000 URIs using the "Carbon Date" tool:

<http://ws-dl.blogspot.com/2013/04/2013-04-19-carbon-dating-web.html>

Note: you'll have better luck downloading and installing the tool rather than using the web service (which will run slowly and likely be unreliable).

For URIs that have greater than Zero Mementos and an estimated creation date, create a graph with age (in days) on one axis and number of mementos on the other.

Listing 6: CarbonDating URIs

```
import json
from ordereddict import OrderedDict
import simplejson

5 import re

from getBitly import getBitlyCreationDate
from getArchives import getArchivesCreationDate
from getGoogle import getGoogleCreationDate
10 from getBacklinks import *
from getLowest import getLowest
from getLastModified import getLastModifiedDate
from getTopsyScraper import getTopsyCreationDate
from htmlMessages import *
15 from pprint import pprint

from threading import Thread
import Queue
import datetime

20 import sys, traceback

fh = open("NonDup.txt", 'r')

25 for line in fh:
    url=line
    url=url.replace('\n', '')

30

    def cd(url):
        #print 'Getting Creation dates for: ' + url

        threads = []
        outputArray = ['', '', '', '', '', '']
        now0 = datetime.datetime.now()

        lastmodifiedThread = Thread(target=getLastModifiedDate, args=(url,
40         outputArray, 0))
        bitlyThread = Thread(target=getBitlyCreationDate, args=(url, outputArray, 1))
        googleThread = Thread(target=getGoogleCreationDate, args=(url, outputArray,
        2))
```

```
archivesThread = Thread(target=getArchivesCreationDate, args=(url,
    outputArray, 3))
backlinkThread = Thread(target=getBacklinksFirstAppearanceDates, args=(url,
    outputArray, 4))
topsyThread = Thread(target=getTopsyCreationDate, args=(url, outputArray, 5))

# Add threads to thread list
threads.append(lastmodifiedThread)
threads.append(bitlyThread)
threads.append(googleThread)
threads.append(archivesThread)
threads.append(backlinkThread)
threads.append(topsyThread)

# Start new Threads
lastmodifiedThread.start()
bitlyThread.start()
googleThread.start()
archivesThread.start()
backlinkThread.start()
topsyThread.start()

# Wait for all threads to complete
for t in threads:
    t.join()

# For threads
lastmodified = outputArray[0]
bitly = outputArray[1]
google = outputArray[2]
archives = outputArray[3]
backlink = outputArray[4]
topsy = outputArray[5]

#note that archives["Earliest"] = archives[0][1]
try:
    lowest = getLowest([lastmodified, bitly, google, archives[0][1], backlink,
        topsy]) #for thread
except:
    print sys.exc_type, sys.exc_value , sys.exc_traceback

result = []

result.append(("URI", url))
result.append(("Estimated Creation Date", lowest))
result.append(("Last Modified", lastmodified))
result.append(("Bitly.com", bitly))
result.append(("Topsy.com", topsy))
```

```

    result.append(("Backlinks", backlink))
    result.append(("Google.com", google))
    result.append(("Archives", archives))
    values = OrderedDict(result)
    r = json.dumps(values, sort_keys=False, indent=2, separators=(',', ': '))

    now1 = datetime.datetime.now() - now0

    #print "runtime in seconds: "
    #print now1.seconds
    #print r
    #print 'runtime in seconds: ' + str(now1.seconds) + '\n' + r + '\n'
    k = str(now1.seconds) + '\n' + r

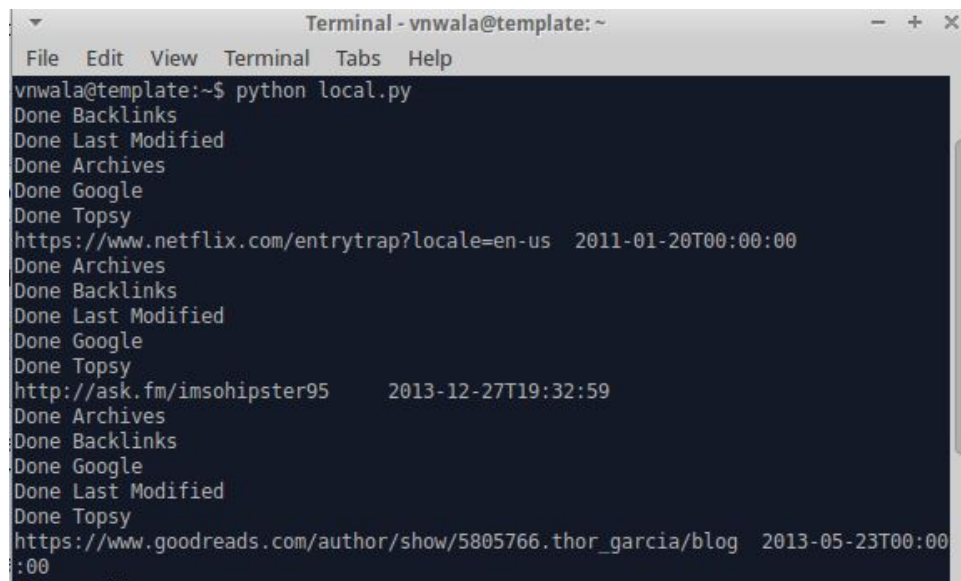
    i = url + "\t" + lowest
    print i
    saveFile = open("carbonDate.txt", 'a')
    saveFile.write(i)
    saveFile.write('\n')
    saveFile.close()
    return r
cd(url)

# if len(sys.argv) == 1:
#     print "Usage: ", sys.argv[0] + " url (e.g: " + sys.argv[0] + " http://www.cs.odu.edu)"
# elif len(sys.argv) == 2:
#     #fix for none-thread safe strptime
#     #If time.strptime is used before starting the threads, then no exception is raised
#     # (the issue may thus come from #strptime.py not being imported in a thread safe
#     # manner). -- http://bugs.python.org/issue7980
#     time.strptime("1995-01-01T12:00:00", '%Y-%m-%dT%H:%M:%S')
#     cd(sys.argv[1])

```

I modified the code local.py given to use to perform this task, collecting only the variables I needed. I also wrote a script to convert the datetime variable to days so that I could construct my plot.

I had only 53 URIs with Memento greater than zero and also had Estimated creation date. Hence 53 points to plot a graph



```
Terminal - vnwala@template: ~  
File Edit View Terminal Tabs Help  
vnwala@template:~$ python local.py  
Done Backlinks  
Done Last Modified  
Done Archives  
Done Google  
Done Topsy  
https://www.netflix.com/entrytrap?locale=en-us 2011-01-20T00:00:00  
Done Archives  
Done Backlinks  
Done Last Modified  
Done Google  
Done Topsy  
http://ask.fm/imsohipster95 2013-12-27T19:32:59  
Done Archives  
Done Backlinks  
Done Google  
Done Last Modified  
Done Topsy  
https://www.goodreads.com/author/show/5805766.thor_garcia/blog 2013-05-23T00:00:00  
Done Topsy
```

Figure 7: Screen Shot of local.py at work

Listing 7: Python script to calculate the Age of URIs

```
import time
import datetime
import calendar

5 fh = open("time.txt", 'r')

for line in fh:
    date=line
    try:
10         def getLowest(date):
            date = date.split(" ")
            date[-1] = date[-1][:18]
            date = " ".join(date)
            epoch = int(calendar.timegm(time.strptime(date, '%Y-%m-%dT%H:%M:%S')))

15             t2 = int(time.time())-epoch
            day = (t2/86400)
            return day
        k = getLowest(date)
20         print k
        saveFile = open("day.txt", 'a')
        saveFile.write(str(k))
        saveFile.write('\n')
        saveFile.close()
25     except BaseException, e:
        print e

fh.close()
```


URI	Estimated Creation Date	Memento	Age in Days (till 09-23)
https://www.netflix.com/entrytrap?locale=en-us	2011-01-20T00:00:00	19	1343
http://newsdigg.net/celebwatcl/lindsay_lohan/rss/	2014-02-08T14:27:34	3	227
http://www.imdb.com/name/nm0000248/	2004-07-14T02:59:47	88	3724
http://www.sweetnsourdeals.com	2012-04-23T16:15:35	92	883
http://suckmydick.com	1999-01-25T04:21:51	235	5721
https://twitter.com/eyoung0816	2012-07-20T00:00:00	7	796
http://newsdigg.net/celebwatcl/leonardo_dicaprio/rss/	2013-06-30T00:00:00	3	451
http://vkidiz.ru	2011-03-11T00:00:00	62	1293
http://www.mindyourownbusiness.com	1999-01-14T19:11:18	417	5731
http://www.onedirectionmusic.com/	2010-09-05T00:00:00	249	1480
https://www.youtube.com/watch?v=hms0ngy4u_w	2008-06-23T04:24:23	71	2284
http://terryborder.com/	2004-06-19T02:40:34	109	3749
http://aliendovecote.com	2009-11-23T00:00:00	32	1760
http://starpittsburgh.cbslocal.com/category/shows/scott-alexander/	2014-05-13T03:49:43	2	134
http://www.subteller.com	2013-04-19T00:00:00	17	523
http://tooclusive.com/	2011-07-27T16:30:01	146	1154
http://www.podbean.com/site/error	2007-05-17T00:00:00	357	2687
http://www.youtube.com/user/iadorewomen1	2011-10-12T06:09:34	11	1078
http://karabomokoko.tumblr.com	2012-10-31T18:44:13	8	632
https://www.youtube.com/watch?v=at1ws18nxs	2008-06-23T08:24:20	71	2283
http://www.cracked.com	2009-05-14T10:59:48	5291	1958
http://illalw aysbesixfeetunderthestars.tumblr.com/	2011-12-18T10:35:58	6	1010
http://www.youtube.com/?app=desktop	2011-04-01T15:15:49	2692	1271
https://twitter.com/sikhyaent	2014-08-18T17:19:27	2	36
http://www.askaboutme.com	2004-12-15T02:08:56	13	3570
http://iqbaleh.tumblr.com	2001-02-01T00:00:00	72	4983
http://www.cityoftyler.org	2002-10-01T00:00:00	365	4376
https://twitter.com/louis_tomlinson/status/120620074301267968	2011-07-22T00:00:00	1	1160
https://twitter.com/	2009-01-12T05:38:55	68206	2081
http://classic.vb-faq.de/	2001-02-01T00:00:00	52	4983
http://www.cinemalaya.org/	2006-07-16T00:00:00	123	2992
http://mentalitch.com	2004-09-23T21:43:07	15	3652
http://some-grace.tumblr.com	2012-11-17T00:00:00	1	676
http://thenextweb.com	2005-11-14T08:01:50	3238	3235
http://www.gamersspin.com	2012-11-21T21:42:40	17	671
http://www.denofgeek.us/	2012-11-24T03:19:51	196	669
http://bigtop.com/	2001-02-01T00:00:00	116	4983
http://www.peta.org/	2000-11-10T00:27:51	4053	5066
http://shopkingcouteure.bigoartel.com	2012-06-30T00:00:00	3	816
https://www.nsa.gov/	2006-02-06T00:00:00	2469	3152
http://www.powder.com/	2000-03-02T11:41:19	336	5318
http://www.palaparealty.com	2009-10-05T03:28:14	7	1815
http://perkstobeinginfinte.tumblr.com/	2013-01-20T08:15:56	7	611
http://beautyforashes61.com	2013-03-19T00:00:00	3	554
http://becomingmums.co.uk	2012-12-04T00:00:00	19	659
http://www.topadulgames.com	2002-02-01T00:00:00	10	4618
https://vine.co/v/m2jtdxwvvd	2014-07-26T00:00:00	1	60
http://www.twentyonepilots.com	2009-12-06T00:00:00	73	1753
http://makeupeverything.com	2012-06-30T00:00:00	4	816
http://www.soundschristmas.com	2002-11-24T01:47:44	54	4322
http://yourwowmoney.com	2012-12-27T00:00:00	4	636
http://www.ocbvibez.blogspot.com	2014-07-23T00:00:00	5	63
http://www.onedirectionmusic.com	2010-09-05T00:00:00	249	1480

Figure 8: Screen Shot Mementos and age in Days

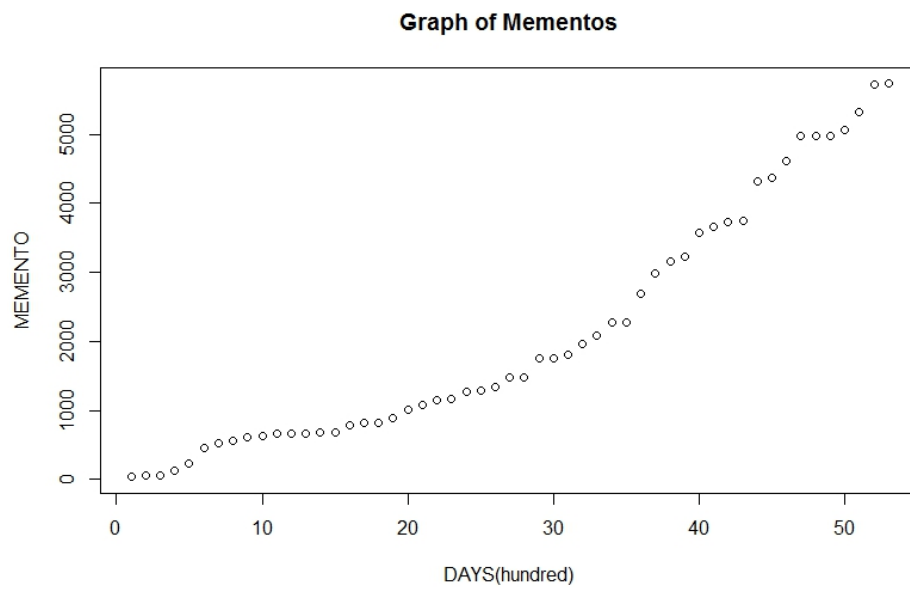


Figure 9: Mementos Vs Age in Hundred Days

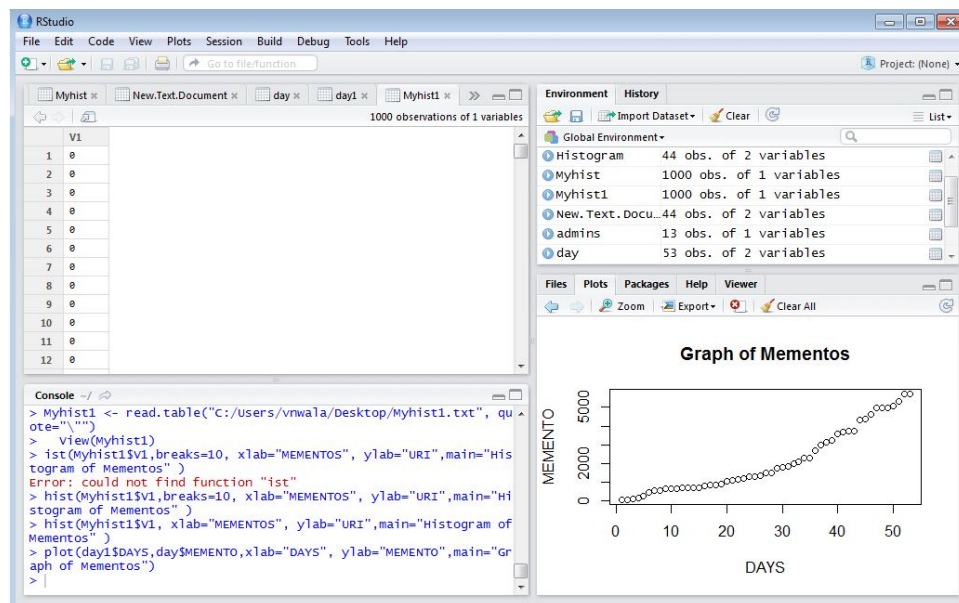


Figure 10: Screen Shot of RgraphLayout

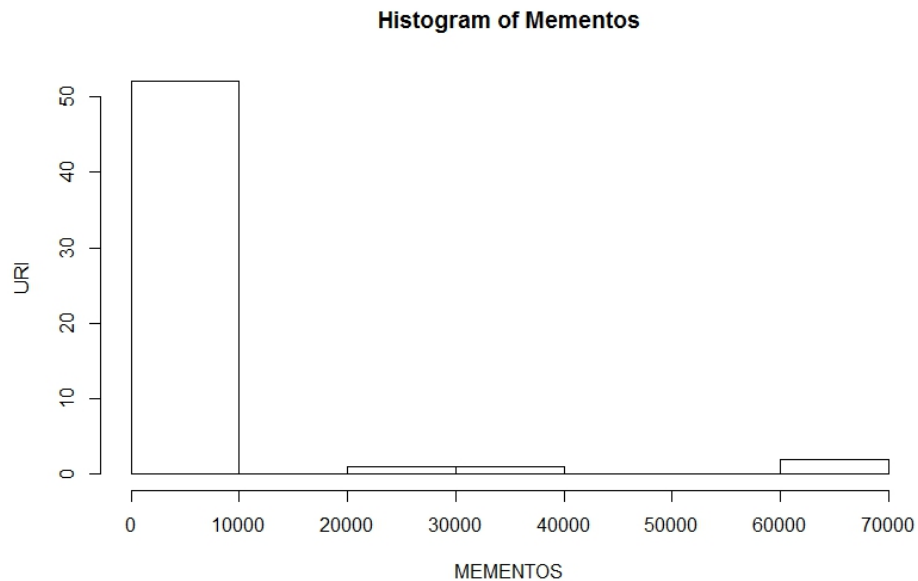


Figure 11: Screen Shot of Histogram without Mementos equal to Zero

OBSERVATIONS AND CONCLUSIONS

Firstly I will like to say that the URLs were randomly selected, some of which might be obscene sites or morally unworthy, there was no intent to be offensive.

Secondly I noticed that my Histogram did not come out well because of the noisy data. I decided to remove all values of Memento equal to zero and plot another histogram and got the result above.

Thirdly I submitted only my final versions of text files, hence they might have different filenames from that seen in my program code.